Robotics

(Course viewgraphs)

João Silva Sequeira[1]

[1]joao.silva.sequeira@tecnico.ulisboa.pt

Winter P2, 2024-2025

Structure of the dynamic equations of a robot

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) \quad (1)$$

- τ ↓ Vector with the forces/torques acting on joint $(n \times 1)$
- $M(\theta)\ddot{\theta}$ ↓ Mass matrix $(n \times n)$
- $V(\theta,\dot{\theta})$ ↓ Vector of centrifugal and Coriolis terms $(n \times 1)$
- $G(\theta)$ ↓ Vector of gravity related terms $(n \times 1)$

Multiplying both sides of (1) by J^{-1} (the pseudo-inverse of the Jacobian) one gets
$J^{-T}\tau = J^{-T}M\ddot{\theta} + J^{-T}V + J^{-T}G$, and using
$F = J^{-T}\tau$, $\dot{x} = J\dot{\theta}$, $\ddot{x} = \dot{J}\dot{\theta} + J\ddot{\theta}$, results in

$$F = J^{-T}M\left(J^{-1}\ddot{x} - J^{-1}\dot{J}\dot{\theta}\right) + J^{-T}V + J^{-T}G = \underbrace{J^{-T}MJ^{-1}}_{M_x(\theta)}\ddot{x} + \underbrace{-J^{-T}MJ^{-1}\dot{J}\dot{\theta} + J^{-T}V}_{V_x(\theta,\dot{\theta})} + \underbrace{J^{-T}G}_{G_x(\theta)}$$

the dynamics equation written in the cartesian space $F = M_x(\theta)\ddot{x} + V_x(\theta,\dot{\theta}) + G_x(\theta)$

Mass distribution I

- Different users have different weights and mass distributions
- Optimize the use of energy sources according to the mass distribution of the user

Mass distribution II

- The mule can carry very different loads
- Optimize energy use

Mass distribution – Inertia tensor

The inertia tensor in frame $\mathcal{F}_A$ is a matrix

$$
{}^{A}I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}
$$

$$
\left.\begin{aligned}
I_{xx} &= \int\int\int_V (y^2 + z^2)\rho \mathrm{d}V \\
I_{yy} &= \int\int\int_V (x^2 + z^2)\rho \mathrm{d}V \\
I_{zz} &= \int\int\int_V (x^2 + y^2)\rho \mathrm{d}V
\end{aligned}\right\} \quad \text{Inertia moments}
$$

$$
\left.\begin{aligned}
I_{xy} &= \int\int\int_V xy\rho \mathrm{d}V \\
I_{xz} &= \int\int\int_V xz\rho \mathrm{d}V \\
I_{yz} &= \int\int\int_V yz\rho \mathrm{d}V
\end{aligned}\right\} \quad \text{Inertia produts}
$$

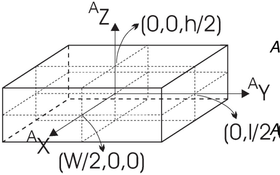
where ρ is the density of the material forming the body

The point mass assumption is often used in robotics, meaning that all the mass is concentrated at the CM

Under the point mass assumption, the inertia matrix written in a frame with origin at the CM is a null matrix

Mass distribution – Inertia tensor – Examples

In the frame of the Center of Mass (CM),



$$^{A}I_{zz} = \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}\int_{-w/2}^{w/2}\rho(x^2+y^2)dxdydz = \rho\frac{wlh}{12}(w^2+l^2)$$

$$^{A}I_{yy} = \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}\int_{-w/2}^{w/2}\rho(x^2+z^2)dxdydz = \rho\frac{wlh}{12}(w^2+h^2)$$

$$^{A}I_{xx} = \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}\int_{-w/2}^{w/2}\rho(y^2+z^2)dxdydz = \rho\frac{wlh}{12}(h^2+l^2)$$

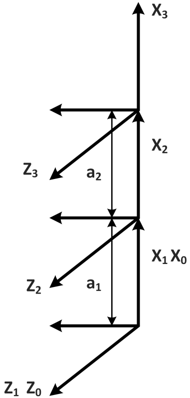
$$^{A}I_{xy} = \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}\int_{-w/2}^{w/2}\rho xydxdydz = 0$$

$$^{A}I_{xz} = \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}\int_{-w/2}^{w/2}\rho xzdxdydz = 0$$

$$^{A}I_{yz} = \int_{-h/2}^{h/2}\int_{-l/2}^{l/2}\int_{-w/2}^{w/2}\rho yzdxdydz = 0$$

^{A}I depends on the reference frame where it is computed

Unconstrained robot dynamics – Lagrangian formulation – Example 1 I



Manipulator RRR planar
(horizontal plane)

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	a_1	0	θ_2
3	0	a_2	0	0

- $\mathcal{F}_3$ only defines the e.e.
- The CM of each link is located at the extremity of the link
- The mass of each link is a point (located at the CM)

Unconstrained robot dynamics – Lagrangian formulation – Example 1 II

$$
{}^0_3T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} c_{12} & -s_{12} & 0 & c_{12}a_2 + c_1 a_1 \\ s_{12} & c_{12} & 0 & s_{12}a_2 + s_1 a_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Potential energy is constant (the manipulator moves in the horizontal plane) and hence it does not affect the dynamic model

$$
L = \sum_{i=1}^{2} \left(\frac{1}{2} m_1 {}^0v_{c_i}^T \, {}^0v_{c_i} + \frac{1}{2} {}^0\omega_i^T \, I_{c_i}^0 \omega_i \right) = \sum_{i=1}^{2} \left(\frac{1}{2} m_1 {}^0v_{c_i}^T \, {}^0v_{c_i} \right) \qquad \text{since } I_{c_i} = \mathbf{0}
$$

Unconstrained robot dynamics – Lagrangian formulation – Example 1

Assuming that CM_2 is at ${}^2P_{c_2} = [a_2, 0, 0]'$ then ${}^0v_{c_2} \equiv {}^0v_3$

From the direct kinematics the origin of $\mathcal{F}_3$ is

$$
{}^0v_{c_2} = \begin{bmatrix} (-s_{12}a_2 - s_1 a_1)\,\dot{\theta}_1 - s_{12}a_2\dot{\theta}_2 \\ (c_{12}a_2 + c_1 a_1)\,\dot{\theta}_1 + c_{12}a_2\dot{\theta}_2 \\ 0 \end{bmatrix}
$$

Similarly, if CM_1 is at ${}^1P_{c_1} = [a_1, 0, 0]'$ then ${}^0v_{c_1} \equiv {}^0v_1$ ${}^0v_{c_1} = \begin{bmatrix} -s_1 a_1\dot{\theta}_1 \\ c_1 a_1\dot{\theta}_1 \\ 0 \end{bmatrix}$

Coming for the kinetic energies $K_1 = \frac{1}{2}m_1 a_1^2\dot{\theta}_1^2$

$$
K_2 = \frac{1}{2}m_2\left(a_2^2\dot{\theta}_2^2 + (a_2^2 + a_1^2 + 2c_2 a_1 a_2)\,\dot{\theta}_1^2 + 2\left(a_2^2 + c_2 a_1 a_2\right)\dot{\theta}_1\dot{\theta}_2\right)
$$

$$
L = K_1 + K_2
$$

Unconstrained robot dynamics – Lagrangian formulation – Example 1

The terms in the dynamics equation are

$$\frac{\partial L}{\partial \theta} = \begin{bmatrix} 0 \\ -2s_2 a_1 a_2 \dot{\theta}_1^2 - 2s_2 a_1 a_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix}$$

$$\frac{\partial L}{\partial \dot{\theta}} = \begin{bmatrix} m_1 \dot{\theta}_1 a_1^2 + m_2 \left(a_2^2 + a_1^2 + 2c_2 a_1 a_2\right) \dot{\theta}_1 + m_2 \left(a_2^2 + c_2 a_1 a_2\right) \dot{\theta}_2 \\ m_2 a_2^2 \dot{\theta}_2 + m_2 \left(a_2^2 + c_2 a_1 a_2\right) \dot{\theta}_1 \end{bmatrix}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = \begin{bmatrix} m_1 a_1^2 \ddot{\theta}_1 - 2m_1 s_2 a_1 a_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 \left(a_2^2 + a_1^2 + 2c_2 a_1 a_2\right) \ddot{\theta}_1 - \\ -m_2 s_2 a_1 a_2 \dot{\theta}_2^2 + m_2 \left(a_2^2 + c_2 a_1 a_2\right) \ddot{\theta}_2 \\ m_2 a_2^2 \ddot{\theta}_2 - m_2 s_2 a_1 a_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 \left(a_2^2 + c_2 a_1 a_2\right) \ddot{\theta}_1 \end{bmatrix}$$

Unconstrained robot dynamics – Lagrangian formulation – Example 1 I

$$
\begin{aligned}
\tau_1 &= m_2 a_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right) + m_2 a_1 a_2 c_2 \left(2\ddot{\theta}_1 + \ddot{\theta}_2\right) + (m_1 + m_2)\, a_1^2 \ddot{\theta}_1 \\
&\quad - m_2 a_1 a_2 s_2 \dot{\theta}_2^2 - 2 m_2 a_1 a_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \\
\tau_2 &= m_2 a_1 a_2 c_2 \ddot{\theta}_1 + m_2 a_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right) - m_2 a_1 a_2 s_2 \dot{\theta}_1 \dot{\theta}_2
\end{aligned}
$$

in vector form

$$
\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} m_2 a_2^2 + 2 m_2 a_1 a_2 c_2 + (m_1 + m_2)\, a_1^2 & m_2 a_2^2 + 2 m_2 a_1 a_2 c_2 \\ m_2 a_2^2 + 2 m_2 a_1 a_2 c_2 & m_2 a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}
$$

$$
- \begin{bmatrix} m_2 a_1 a_2 s_2 \dot{\theta}_2^2 + 2 m_2 a_1 a_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 a_1 a_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix}
$$

TÉCNICO LISBOA

What if the end-effector is constrained to be in contact with a surface ?

Lagrangian formulation – Example – Lara Croft's braid/ponytail

Credits: news.bbc.co.uk/cbbcnews/hi/sci_tech/newsid_2364000/2364099.stm

Credits: www.laracroft.com

Jacobians, Forces, and Torques

Work produced in an infinitesimal displacement

$$\underbrace{\underbrace{F^T \delta x}_{\text{work in the cartesian space}} = \underbrace{\tau^T \delta\theta}_{\text{work in the control space}}}_{\text{work must be independent from the reference frame where it is computed}}$$

Substituting the definition of Jacobian $\delta x = J\delta\theta$ above

$$F^T J\, \delta\theta = \tau^T\, \delta\theta \quad \rightarrow \quad F^T J = \tau^T \quad \rightarrow \quad \tau = J^T F$$

Re-interpretation of the concept of singularity (studied for serial manipulators)

In a singularity, there are forces that do not require the controls to produce any torque

Constrained robot dynamics

- The relation $\tau_f = J(\theta)^T f_{external}$ expresses how the external forces, $f_{external}$, are mapped in torques through the geometry of the robot; this component can be added directly in the dynamics equations

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) = \tau + \tau_f$$

Constrained robot dynamics – Using the Lagrangian-based algorithm

- Let the constraints be of one of the following types

$A(\theta) = 0$	Position constraint
$J(\theta)\dot{\theta} = B(\theta)$	Velocity constraint

- The generalized torques imposed by each of the constraints are given by

$\tau_C = \left[\frac{\partial A(\theta)}{\partial \theta}\right]^T \lambda$	For position constraints
$\tau_C = J(\theta)^T \lambda$	For velocity constraints

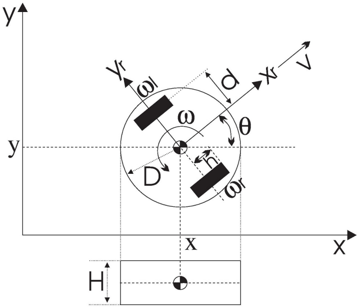
- Solve the resulting differential-algebraic system

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) = \tau + \tau_C,$$

subject to the constraints, for the unknowns (θ, λ)

TÉCNICO LISBOA

Constrained robot dynamics – Lagrangian formulation – Example 2



- Unicycle robot with circular body
- Mass is uniformly distributed
- The free Lagrangian is computed as if there were no constraints

$$L = \frac{1}{2} m \, [v, 0, 0] \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2} \, [0, 0, \omega] \, I_c \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} = \frac{1}{2} m v^2 + \frac{1}{2} I_{c_{zz}} \omega^2 \quad = \quad \frac{1}{2} m v^2 + \frac{1}{12} \frac{H^3 D^3}{144} \pi \omega^2$$

$\frac{\partial L}{\partial \theta} \equiv \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \frac{\partial L}{\partial \dot{\theta}} \equiv \begin{bmatrix} mv \\ I_{c_{zz}} \omega \end{bmatrix} \quad \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) \equiv \begin{bmatrix} m\dot{v} \\ I_{c_{zz}} \dot{\omega} \end{bmatrix}$ and then $\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = \tau \quad \longrightarrow \begin{bmatrix} F \\ N \end{bmatrix} = \begin{bmatrix} m\dot{v} \\ I_{c_{zz}} \dot{\omega} \end{bmatrix}$

Constrained robot dynamics – Lagrangian formulation – Example 2

The unicycle motion constraint is

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0$$

which yields for the main equation

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_{c_{zz}} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ N_z \end{bmatrix} + \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \end{bmatrix} \lambda$$

Constrained robot dynamics – Lagrangian formulation – Example 2

If the goal is to know the torques required at each motor, under no constraints, the free Lagrangian is

$$L = \frac{1}{2} m \left(h \frac{\omega_d + \omega_e}{2} \right)^2 + \frac{1}{2} I_{czz} \left(h \frac{\omega_d - \omega_e}{2d} \right)^2$$

$$\frac{\partial L}{\partial \theta} \equiv \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\frac{\partial L}{\partial \dot\theta} \equiv \begin{bmatrix} \frac{h^2}{2} m \left(\omega_d + \omega_e\right) + \frac{h^2}{2d^2} I_{czz} \left(\omega_d - \omega_e\right) \\ \frac{h^2}{2} m \left(\omega_d + \omega_e\right) - \frac{h^2}{2d^2} I_{czz} \left(\omega_d - \omega_e\right) \end{bmatrix} \qquad \frac{d}{dt} \left(\frac{\partial L}{\partial \dot\theta} \right) \equiv \begin{bmatrix} \frac{h^2}{2} m \left(\dot\omega_d + \dot\omega_e\right) + \frac{h^2}{2d^2} I_{czz} \left(\dot\omega_d - \dot\omega_e\right) \\ \frac{h^2}{2} m \left(\dot\omega_d + \dot\omega_e\right) - \frac{h^2}{2d^2} I_{czz} \left(\dot\omega_d - \dot\omega_e\right) \end{bmatrix}$$

and hence

$$\begin{aligned} N_{\text{right}} &= \frac{h^2}{2} \left(m + \frac{I_{czz}}{d^2} \right) \dot\omega_d + \frac{h^2}{2} \left(m - \frac{I_{czz}}{d^2} \right) \dot\omega_e \\ N_{\text{left}} &= \frac{h^2}{2} \left(m - \frac{I_{czz}}{d^2} \right) \dot\omega_d + \frac{h^2}{2} \left(m + \frac{I_{czz}}{d^2} \right) \dot\omega_e \end{aligned}$$

Constrained robot dynamics – Lagrangian formulation – Example 2

- If the alternative formulation for the kinematic constraint is used (with ρ the curvature radius of the trajectory)

$$\omega_r \left(1 - \frac{\rho}{d}\right) + \omega_l \left(1 + \frac{\rho}{d}\right) = 0$$

- Which is the same as

$$v = \omega\rho$$

- The constraint torques are

$$\tau_c = \begin{bmatrix} 1 - \frac{\rho}{d} \\ 1 + \frac{\rho}{d} \end{bmatrix} \lambda$$

Constrained robot dynamics – Lagrangian formulation – Hint on the solution of constrained dynamics

- Given $M(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) = \tau + \tau_c$ subject to $A(\theta) = 0$

$$A(\theta) = 0$$

$$\dot{A}(\theta)\dot{\theta} = J(\theta)\dot{\theta} = 0$$

$$\ddot{A}(\theta) = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta} = 0$$

$$\ddot{\theta} = M^{-1}\left(-N + \tau + \tau_c\right) = M^{-1}\left(-N + \tau + J^T\lambda\right)$$

- Substituting above

$$JM^{-1}J^T\lambda = -\dot{J}\dot{\theta} - JM^{-1}\left(-N + \tau\right)$$

$$\lambda = \left(JM^{-1}J^T\right)^{-1}\left(-\dot{J}\dot{\theta} - JM^{-1}\left(-N + \tau\right)\right)$$

Constrained robot dynamics – Lagrangian formulation – Hint on the solution of constrained dynamics I

- Assuming that it is possible to compute J^{-1}

$$\ddot{\theta} = J^{-1}\dot{J}\dot{\theta}$$

- If J^{-1} does not exist numerical methods must be used

- For the unicycle

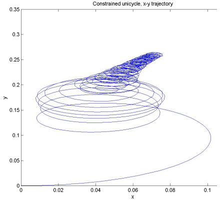
$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_{c_{zz}} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ N_z \end{bmatrix} + \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \end{bmatrix} \lambda$$

- In equilibrium, $\ddot{x} = \ddot{y} = \ddot{\theta} = 0$ and λ only compensates the controls F_x, F_y, N_z

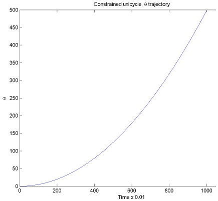
- If the robot wants to move freely through the action of the controls F_x, F_y, N_z then it must compensate for the forces/torques imposed by the kinematic constraint

Constrained robot dynamics – Lagrangian formulation – Hint on the solution of constrained dynamics II

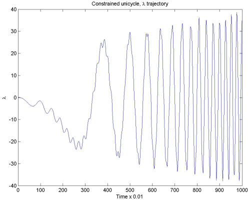
- For a unicycle with mass $m = 1$ Kg, inertia $I_{c_{zz}} = 0.1$ Kg.m^2, and controls $(F_x, F_y, N_z) = (1, 0.1, 1)$.



Trajectory in the plane



Orientation



λ trajectory

- Specifying a control $F_y \neq 0$ results in no sideways motion, as it should be expected for this robot (you should test this :))

- This dynamics ends up by having only 2 dof, F_x, for linear motion, and N, for rotational motion

TÉCNICO LISBOA

Constrained robot dynamics – Lagrangian formulation – Hint on the solution of constrained dynamics III

- An alternative solution to the DAE involves multiplying both sides of the dynamics by a matrix $S(\theta)$ such that

$$S(\theta)\tau_c = 0$$

- That is, $S(\theta)$ must be chosen in the null space of $J^T(\theta)$, i.e., $S(\theta)\,J^T(\theta) = 0$, yielding

$$S(\theta)M(\theta)\ddot{\theta} + S(\theta)N(\theta, \dot{\theta}) = S(\theta)\tau \qquad (2)$$

TÉCNICO LISBOA

Including dissipative effects in robot dynamic models

- Models of dissipative effects are expressed in terms of forces and torques (hence they can be included directly in the dynamic model equations)

- Examples of dissipative torques in manipulators

$$\begin{aligned}
\tau_{\text{viscous}} &= \nu\dot{\theta} \longrightarrow \text{Viscous friction} \\
\tau_{\text{coulomb}} &= c\,\text{sgn}(\dot{\theta}) \longrightarrow \text{Coulomb friction} \\
\tau_{\text{total friction}} &= c\,\text{sgn}(\dot{\theta}) + \nu\dot{\theta} \qquad (3)
\end{aligned}$$

- c and ν are constants depending, for example, of joint lubrication

- Model (3) for $\tau_{\text{total friction}}$ is highly simplified; in general each term depends on the robot configuration

- In "high performance" robots, e.g., boats and airplanes, hydrodynamic/aerodynamic drag forces have to be included; γ is a constant

$$f_{\text{aero}} = \gamma\,\dot{\theta}^2$$

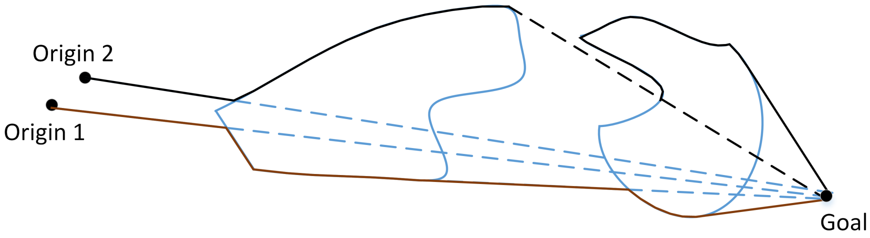
TÉCNICO LISBOA

Path planning vs path tracking vs trajectory following

- Path planning generates a sequence of points representing a sequence of arcs of curve (possibly straight lines) leading from the origin to the goal
- Path tracking aims at moving along the sequence of points in a previously planned path.
- Trajectory following aims at moving as closely as possible to a reference trajectory, e.g., constructed using the points in a planned path

Bug algorithms

- Assumes no info on the environment
- The robot is a point (adapting to a non-point robot is doable) and moves reactively
- Follow in the direction of the goal; if not possible follow the boundary of the obstacle until there's enough free space to go straight to the goal



The pure pursuit algorithm

- Path tracking algorithm
- A look-ahead point moves along the path
- The robot must pursue the look-ahead points
- It requires the control of the look-ahead point plus the control of the robot itself